The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Human Augmented Mapping for Indoor Environments using a Stereo Camera

Soohwan Kim, Howon Cheong, Ju-Hong Park, and Sung-Kee Park

Abstract— **In this paper, we suggest a new method of human augmented mapping for indoor environments using only a stereo camera. Through user's help, a robot with a stereo camera can investigate the environment without failure and even more efficiently. Moreover, the user can share the information about the environment with the robot and add semantic information to the environmental map. We employ PCA features for visual landmarks and a hybrid map for map representation. Particularly, we define two types of nodes, U/R-nodes and divide the map building into three processes, User's Guidance, Robot's Map Revision, and Robot's Map Completion. We implemented a human augmented mapping system with a stereo camera and demonstrated it in rectangular-shaped corridors. From the comparison with a manually-built map, we showed the feasibility of the environmental map generated by our proposed method.**

I. INTRODUCTION

Map building is a prerequisite step for mobile robot localization and navigation. Over the past several decades many researches have been devoted on autonomous map building such as SLAM [1] [2] and exploration [3]. Practical results, however, have been made with range sensors like laser range finders and sonar sensors. The reason there is no fully automatic map building system using only vision is because passive cameras cannot obtain complete depth information in common environments which usually include textureless objects.

Recently, human augmented mapping [5] has been introduced for semi-autonomous map building. It addresses the problems of vision-based autonomous map building that it is difficult to carry out exploration in unknown environments due to the incomplete depth information. Moreover, the main purposes of automatically generated maps are for localization and navigation, not for services. Thus, the user has to understand the map and add sematic information like place labeling for user-friendly services.

The main concept of human augmented mapping is that a user guides his or her mobile robot through the environment, while the robot interacts with the user and builds the map. By doing that, the robot can build the environmental map efficiently, and the user can share the knowledge about the environment with the robot. Another advantage of this approach is that it makes vision-based map building possible; only with cameras the robot may not decide where to go or revisit the same place more than once. Through the user's help, however, those problems can be resolved and it can even investigate user's favorite places precisely.

P. Althaus and H. I. Christensen [4] proposed a semi-autonomous map building system for domestic environments with a laser scanner and sonar sensors. They employed topological map representation of which nodes are necessary for the robot to navigate the environment like a corridor, room, and door. While the robot follows the user with a laser scanner, it builds the map online by extracting doorways and corridors from sonar data. However, the system is informed through a wireless laptop whenever the user leaves a room or enters a corridor.

E. A. Topp and H. I. Christensen [5] suggested a tracking method for following and passing persons. Particularly, they used the expression of ”Human Augmented Mapping” for the first time in that paper and developed a human tracking system with a laser scanner.

Albert Diosi et al. [7] proposed an interactive SLAM method using a laser scanner and advanced sonar sensors. The robot builds a occupancy grid map using a Kalman filter framework while following the user. Their work is focused on place segmentation after acquiring the occupancy grid map using SLAM. For that, the user places virtual markers in the map, while guiding the robot, and the robot labels each rooms of the grid map using watershed segmentation and marker-guided merging.

E. A. Topp and H. I. Christensen [6] enhanced their previous researches [4] [5] and implemented a robot system for human augmented topological mapping with a laser range finder. In that paper, they suggested a general framework of human augmented mapping which includes two types of events, external input from the user and internal detection from the robot. Particularly, they defined a region descriptor with the mass and the maximum range along the two principle components of the laser data and applied it for a classification or categorization approach to facilitate localization.

The novelty of our approach is to utilize only a stereo camera for human augmented mapping, while registering user's interesting places with user's minimum help. Thus, without accurate range sensors like laser range finders the robot can build the environmental map efficiently through

This work was supported by the Intelligent Robotics Development Project of the 21st Century Frontier R&D Programs of the Ministry of Knowledge and Economy, Republic of Korea.

Soohwan Kim is a research scientist of Korea Institute of Science and Technology, kimsoohwan@kist.re.kr

Howon Cheong is a PhD student of Korea Institute of Science and Technology, dest2@kist.re.kr

Ju-Hong Park is an engineer of MtekVision Co, Ltd., 3F, World Meridian Venture Center II, 426-5, Gasan-Dong, Geumchun-Gu, Seoul, Republic of Korea, parkjh@mtekvision.com

Sung-Kee Park is a principal research scientist of Korea Institute of Science and Technology, 39-1 Hawolgok-Dong, Wolsong-Gil 5, Seongbuk-Gu, Seoul, Republic of Korea, skee@kist.re.kr

978-1-4244-3804-4/09/$25.00 ©2009 IEEE

following the user. Moreover, our approach solves map building and place labeling problems at the same time. We employ PCA features [9] as visual landmarks and a hybrid map as map representation. Particularly, we define two kinds of nodes, U/R-nodes and divide map building into three processes, User's Guidance, Robot's Map Revision, and Robot's Map Completion. We also propose two-way pose estimation to revise the erroneous global topological map which is based on the incorrect odometry.

II. SYSTEM OVERVIEW

A. Robot System

In this paper, we only use a stereo camera as the sensor modality. With other sensors like a laser range finder, we might be able to build more accurate occupancy grid maps, or assign each sensor to a specific function such as the stereo camera for human following and the laser range finder for map building, or vice versa [7].

However, we restrict the sensor modality to a stereo camera for lowering the robot's price and making the robot system as simple as possible. Thus, our robot applies the stereo camera for both human robot interaction and map building, which means it cannot build a map, while interacting with the user.

Fig 1. shows the mobile robot used in this paper. The stereo camera is mounted on a pan/tilt module in front of the laptop computer which displays the status of the robot. Note that the sonar sensors attached to the robot are not used in this paper. For your information, we do not actually use the panning module, since it was not calibrated at that time. Instead, the robot turns its body to look around.

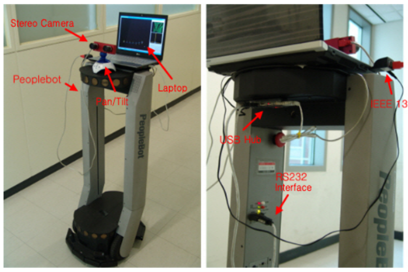

Fig. 1. Our mobile robot system

B. Experimental Environment

In this paper, we only consider indoor environments like corridors of office buildings. Fig 2. shows rectangular-shaped corridors used in the experiments of which size is about 17m × 10m. Note that the left side of the corridor is covered with glass walls so that reflected and transmitted images make vision-based exploration highly challenging. Moreover, since the geometries of corridors are too simple and the glass walls are transparent, a robot even with laser range finders hardly succeeds in localization or navigation in those environments. In other words, for those environments which are quite common but too difficult to navigate, our vision-based human augmented mapping approach is appropriate.

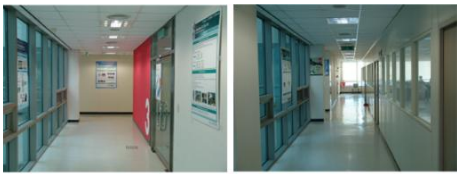

Fig. 2. The experimental environment, corridors

C. Map Building Process

In this paper, we assume that users have no background knowledge about robotics. Thus, the mobile robot should provide users with easy interaction methods and ask them as little helps as possible for map building. For that reason, we divide human augmented mapping into three processes.

1) User's Guidance: The user guides the robot through the environment and tells interesting places which are required for user-friendly services.

2) Robot's Map Revision: The robot revises the erroneous environmental map with accumulated data which is obtained while following the user.

3) Robot's Map Completion: The robot autonomously follows the trajectory again and revisits some places to acquire environmental data necessary for navigation and localization.

Of course, three processes can be done at the same time. But in that case, the user has to wait for the robot to obtain environmental data even in non-interesting places, which may bother naive users. However, with proposed separate processes, all the user has to do is to guide the mobile robot to favorite places in the first process, User's Guidance; anything else will be taken care of by the robot autonomously.

III. VISUAL LANDMARKS AND MAP REPRESENTATION

Map building is strongly connected with localization and navigation, since the latter utilizes the result of the former. In that sense, this paper is based on our previous work for vision-based global localization [8] where we used manually generated environmental maps. However, this section is more than the summary of our previous work about visual landmarks and map representation; we define two types of nodes, U-node and R-node which is designed to minimize user's help while making the robot to build useful environmental maps.

A. Visual Landmarks

In vision-based localization, a mobile robot matches visual features in the current view with models in the environmental maps and estimates its pose with matched ones. Here, we employ PCA features [9] for visual landmarks which is known to be suitable for vision-based localization due to its robustness to scale, rotation and small amount of illumination changes.

By the way, since PCA features are extracted from images, to localize the robot's pose, we need to associate PCA features with positional data in the three dimensional space. Fortunately, you can obtain the 3D position of each pixel in the camera coordinates directly through APIs of commercial stereo cameras. In this paper, we just combine PCA features with their 3D positions and call them 3D PCA features for short.

B. Map Representation

Map representation tells about how to store visual landmarks extracted from the environment. We employ hybrid map representation consisting of a global topological map and local metric maps [8].

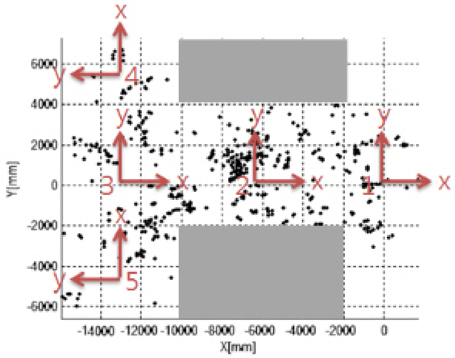

Fig. 3. An example of a hybrid map, bird's eye view

Fig 3. shows an example of a hybrid map. In the global topological map, the connectivity between five nodes (i.e., 1-2, 2-3, 3-4, and 3-5) and the relative poses between connected nodes are recorded. In each node's local metric map, 3D PCA features are expressed in each node coordinates. Note that the black points indicate 3D PCA features of each node, and they are transformed with respect to the origin of Node 1 in order to draw them in one picture.

The reason we employ hybrid map representation for human augmented mapping is, first of all, because it is similar with the way human beings do [10]. Secondly, it is more appropriate to vision-based localization than a global metric map. Since PCA features are basically extracted from images, the descriptors of two PCA features in the same position could be highly different from various views and distances. Thus, it is better to express 3D PCA features in the node coordinates where they are captured. Moreover, since merging local metric maps into one global metric map causes accumulated errors, the accuracy of the robot's pose in a global metric map decreases as the robot goes far from the origin.

C. U/R-nodes

As mentioned above, all the user has to do in our map building scenario is to bring the mobile robot where he or she wants the robot to remember. More than that would be uncomfortable and unnecessary for naive users. However, if the user registers scattered regions in large scale environments so that local maps are not overlapped, it is difficult for the robot to localize or navigate with that map. In other words, additional data between two long-distanced nodes are needed like stepping stones.

However, it does not match with our main concept that the user should consider the distance between two nodes and intentionally register unnecessary places between them, or the robot is supposed to warn the user to add a new node, whenever it moves too far from the lastly registered place.

Therefore, in this paper we define two types of nodes, U-node and R-node. U-nodes are the regions where the user registers to the robot in the User's Guidance process. When the robot is commanded to add a new U-node, it looks around to capture omnidirectional images and build a local map by extracting 3D PCA features from captured images. At the same time, the user can label a U-node and attach semantic information to it for user-friendly services on the future. By doing that, the user can share the information about the environment with his or her robot.

On the other hand, R-nodes are the places on the trajectory which are inserted automatically by the robot for localization and navigation. The user does not have to know about R-nodes, because it is meaningless for the user. Note that U/R-nodes are defined in the User's Guidance process, and there is only a front view image in R-nodes as environmental data. That is why the robot should revisit R-nodes in the Robot's Map Completion process. How the robot determines additional R-nodes on the trajectory and how it revisits them without the user's help will be described in section VI.

IV. USERS'S GUIDANCE PROCESS

In this process, the user guides the mobile robot just like you do when your guest visits your home. For that, we constructed four commands: FOLLOW, CREATE-A-NODE-THERE, CREATE-A-NODE-HERE, and QUIT. Each command is inputted through virtual buttons on the screen. Since you can use other methods for human robot interaction like wireless laptops or verbal commands, we omit the details about virtual buttons here.

Fig 4. shows the overall flowchart for the User's Guidance process. The details will be explained in the following subsections.

A. Human Following

If the user commands FOLLOW, the robot recognizes the face of the user and keeps the distance (in this paper, 2m) between the robot and the user while following him or her. We also implemented a tracking system so that the robot can follow the user, even though he or she turns back and goes forward. The implementation details of the face recognition and tracking system is omitted due to the scope of this paper. Of course, you can use remote controllers like a joystick or keyboard for human following.

Note that while the robot follows the user, it saves some data (a front view image, 3D positional data of each pixel in the front view, and odometry data) regularly (in this paper, every movement of 30cm and every turn of 10 degrees). Those will be used in the next process to revise the errors in

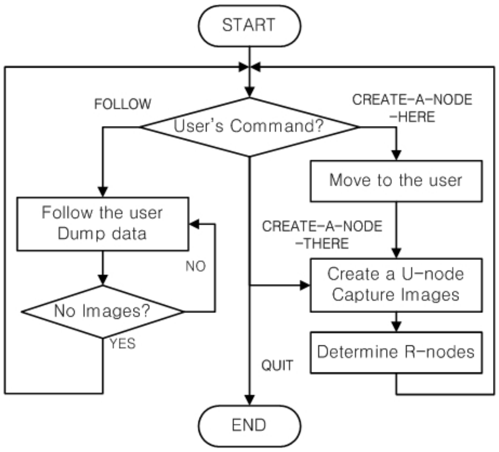

Fig. 4. Overall flowchart of the User's Guidance process

the global topological map which is caused by the incorrect odometer.

Of course, at the center of the front view there must be the user, which is not good for extracting enough visual landmarks from the image. Thus, we tilt the stereo camera 15 degrees up to reduce the portion of the user in the robot's view.

B. Creating a U-node

When the mobile robot arrives at the user's favorite place, the user blocks the camera to stop it. He or she can command CREATE-A-NODE-THERE or CREATE-A-NODE-HERE to register a U-node at the robot's position or at the user's one, respectively. Sometimes there could be some places where it is difficult to bring the robot like corner points. The latter is designed for those cases.

At this time, the user can entitle the node and attach semantic information for user friendly services. The robot obtains the current pose from the odometer and records the connectivity and relative pose in the global topological map. For the local metric map of the current U-node, it looks around and takes omnidirectional images(in this paper, 8 images every 45 degrees) to extract visual landmarks from the environment.

C. Determining R-nodes

After registering a U-node, the robot determines where to add R-nodes for localization and navigation. If the distance between two nodes is too far, or one node is not visible from the other, it is hard to localize and navigate between them. Thus, in this paper the robot divides the trajectory equally to make the distance between two nodes 2m ∼ 3m and inserts R-nodes at those joints. Also, whenever it turns more than 20 degrees, a R-node is inserted. Here, in order to revise the global topological map in the next process, R-nodes are inserted at the points where the robot dumped environmental data while following the user.

However, R-nodes in this process is empty; their positions are recorded in the global topological map, but there are no local metric maps for them. Therefore, the robot should revisit them and collect omnidirectional images like it did at every U-node. Note that the global topological map in this process is erroneous, since it is based on the odometry data.

V. ROBOT'S MAP REVISION PROCESS

The hybrid map built in the previous process is incorrect and incomplete due to the erroneous odometer. Therefore, in this process the robot revises the erroneous global topological map using dumped data while following the user. We assume that initially the robot made a U-node at the starting point, otherwise a R-node is inserted at the starting point. Since the nodes are connected in the global topological map, we revise the global topological map incrementally, i.e., node by node from the starting point.

In order to compensate the relative pose of two connected nodes, we use dumped data along the trajectory; front view images, 3D positional data of each pixel in the front views, and odometry data. Here, the nodes can be an any type of node, U-node or R-node. The difference is the number of captured images; a U-node has omnidirectional images, but a R-node has only one front view image. Although the image of a R-node is one-directional and even covered by the user, it is still useful for a temporary local metric map.

Now, we apply the particle filter to compensate the relative pose of two connected nodes. Algorithm 1. (reproduced from [1]) describes how to estimate the robot's pose at time t X_t with the previous pose X_{t-1}, the action u_t, and the observation z_t. Since it is a well-known algorithm, we would not explain it in details but how to be applied in this paper.

Algorithm 1 Particle_filter(X_{t-1}, u_t, z_t) from [1]

```
$\overline{X_t} = \emptyset$
**for** $m = 1$ to $M$ **do**
    draw $x_t^m \sim p(x_t|u_t, x_{t-1}^m)$          $\leftarrow$ motion update
    $w_t^m = p(z_t|x_t^m)$                         $\leftarrow$ measurement update
    $\overline{X_t} = \overline{X_t} + \langle x_t^m, w_t^m \rangle$
**end for**
$X_t = \emptyset$
**for** $m = 1$ to $M$ **do**
    draw $x_t^m$ with probability $\propto w_m^t$        $\leftarrow$ resampling
    $X_t = X_t + \langle x_t^i, 1/M \rangle$
**end for**
**return** $X_t$
```

If node i is a U-node, it has a local metric map generated in the previous process. On the other hand, in a R-node the robot can build a temporary local metric map with a front view image. Thus, in any cases, we have a local metric map, odometry data, and dumped data for each step.

Now, given actions (odometry data) $U_{i:j}$ and observations (dumped data) $Z_{i:j}$ of every step from node i and node j, we can calculate the relative pose of node j with respect to node i using Algorithm 2.

In Algorithm 2., we apply the particle filter every step (line 3). After that, the relative pose between two connected nodes can be computed from the weighted pose of all particles (line

Algorithm 2 Pose_estimation$(U_{i:j}, Z_{i:j})$

```
1: randomly sample $X_0$
2: for $t = i$ to $j$ do
3:     $X_t$ =Particle_filter$(X_{t-1}, u_t, z_t)$
4: end for
5: $x = \sum_{m=1}^{M} w_j^m \times x_j^m$        ← weighted pose
6: return transformation matrix ${}^{i}T_j$ of $x$
```

5). Finally, the relative pose of node i respect to node j is returned as a transformation matrix, ${}^{i}T_j$.

Here, we can use that transformation matrix as the revised relative pose. However, the accuracy of the estimated robot's pose goes down as the robot moves from node i to node j, since the map of node i do not perfectly cover to node j. Fortunately, we have two local metric maps for node i and j, which means we can estimate the relative pose in two directions, forward and backward. One may think merging the results of forward and backward pose estimation. But that approach does not eliminate the accuracy degrading problem. Therefore, we propose two-way pose estimation which is described in Algorithm 3.

Algorithm 3 Two-way_pose_estimation$(U_{i:j}, Z_{i:j})$

```
1: $k = [(i + j)/2]$                                   ← middle point
2: ${}^{i}T_k$ = Pose_estimation$(U_{i:k}, Z_{i:k})$     ← forward
3: ${}^{j}T_k$ = Pose_estimation$(U_{j:k}, Z_{j:k})$     ← backward
4: ${}^{i}T_j = {}^{i}T_k \times {}^{k}T_j = {}^{i}T_k \times ({}^{j}T_k)^{-1}$
5: return ${}^{i}T_j$
```

Rather than merging forward (from node i to node j) and backward (from node j to node i) results, we pick the middle point k on the trajectory between two nodes i and j (line 1). And then, we estimate the forward pose from node i to point k, ${}^{i}T_k$ (line 2) and the backward pose from node j to point k, ${}^{j}T_k$ (line 3), respectively. The fusion of two estimated poses to the middle point k produces the relative pose of two nodes ${}^{i}T_j$ (line 4).

The only difference between Algorithm 2. and Algorithm 3. is picking the middle point k, but that increases the accuracy of the relative pose dramatically. By doing this two-way pose estimation iteratively from the first node to the last node, we can revise the global topological map.

VI. ROBOT'S MAP COMPLETION PROCESS

Now, we have a revised topological map, but R-nodes still have insufficient environmental data. Thus, in this final process the robot revisits R-nodes one after another to capture omnidirectional images and build local maps. However, ironically, in order to revisit R-nodes local metric maps of R-nodes are needed first. Therefore, to resolve that dilemma we assume that the robot is placed near the starting point to follow the trajectory again.

Fig 5. shows the overall flowchart for the Robot's Map Completion process.

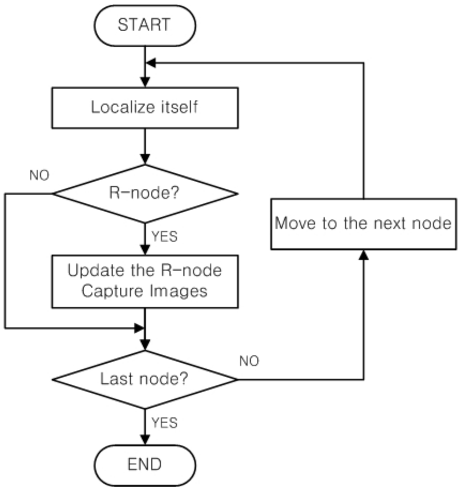

Fig. 5. Overall flowchart of the Robot's Map Completion process

First of all, the robot localizes itself in a node using our previous work on vision-based global localization [8]. If the current node is a R-node, then it captures omnidirectional images and extracts 3D PCA features to build a local metric map. After that, it updates the pose of the current node in the global topological map. This is because the robot may not arrive at the exact desired point due to odometry errors. Note that U-nodes are not updated in this process, since those are shared places with the user so that the robot cannot change its position alone.

The robot moves to the next node and does the same thing until there is no place to visit. Here, the details of path planning and obstacle avoidance is omitted because it is out of the scope of this paper. For that, we used APIs for point-to-point movement in ARIA [11] and experimented in a static environment.

Before moving on to the next node, the robot revises the relative pose of the next node to the current node in the global topological map, which will be explained precisely in the following subsection.

VII. EXPERIMENTAL RESULTS

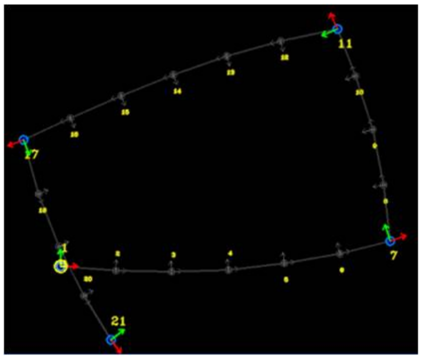

Fig. 6. Global topological map built in the User's Guidance process

Fig 6. shows the erroneous global topological map obtained in the User's Guidance process. We guided our mobile robot counterclockwise along the corridor and registered U-nodes(1, 7, 11, 17, and 21) at the four corners. Note that U-node 1 and U-node 21 are the same place but they are not coincident.

The omnidirectional images of four U-nodes are shown in Fig 7. The 3D PCA features extracted from them are recorded in each node's local metric maps.

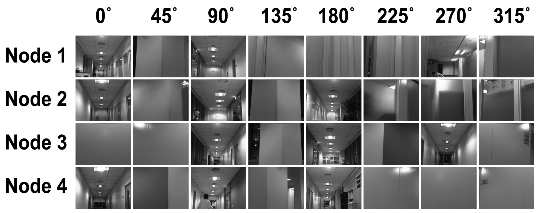

Fig. 7. Omnidirectional images of U-nodes

Fig 8. shows the revised and updated global topological map in the Robot's Map Revision and Completion process. You can see the positions of R-nodes are different from those of Fig 6. This is because the robot cannot revisit R-nodes exactly due to the odometry error.

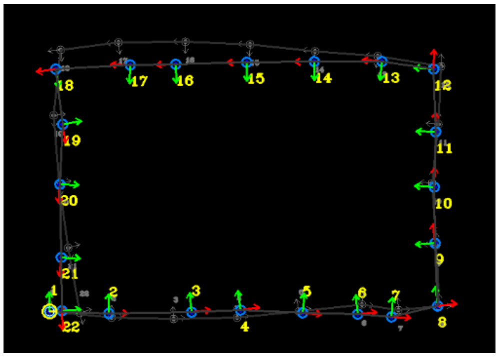

Fig. 8. Global topological map revised and updated in the Robot's Map Revision and Completion process

The robot started from node 1. Because it is a U-node, the robot did not turn around to collect omnidirectional images. After localizing itself, it revised the relative pose of node 2 to node 1 using dumped data. With the point-to-point movement, it progressed to the next node, node 2. Now, since node 2 is a R-node, the robot looked around and built a local metric map. After that, it updated the pose of the R-node with the current pose in the global topological map. That procedure was repeated until the final node.

Note that although we do not employ loop closing, the first and last node (Node 1 and Node 21) are located so closely and the trajectory looks like a rectangle. We measured the position of each node manually. The average positional error is less than 30cm, and the rotational error of nodes is less than 10 degrees, which is almost same with those of manually built hybrid maps in the previous work [8].

VIII. CONCLUSION AND FUTURE WORK

In this paper, we suggest a new method of human augmented mapping for indoor environments using only a stereo camera. Through the user's guidance, our mobile robot with a stereo camera can investigate the environment without failure and even more efficiently. Moreover, the user can share the information about the environment with the robot and add semantic information to the environmental map.

We employ PCA features for visual landmarks and a hybrid map for map representation. In order to minimize user's help, we define two kinds of nodes and separate map building into three processes, User's Guidance, Robot's Map Revision, and Robot's Map Completion.

Particularly, the global topological map is erroneous in the User's Guidance process. Thus, we propose the two-way pose estimation method to fix it in the Robot's Map Revision process. The experimental results show that the environmental map generated by our proposed system is acceptable and feasible for vision-based localization and navigation.

However, in this paper we do not consider branch nodes which is necessary for more complex topological map. Also, we do not apply loop closing when revising the global topological map.

IX. ACKNOWLEDGMENTS

This work was supported by the Intelligent Robotics Development Project of the 21st Century Frontier R&D Programs of the Ministry of Knowledge and Economy, Republic of Korea.

REFERENCES

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, The MIT Press, 2005.

[2] R. Sim, P. Elinas, M. Griffin, and J. J. Little, ”Vision-based SLAM using the Rao-Blackwellised Particle Filter,” *Proc. Of the IJCAI Workshop on Reasoning with Uncertainty in Robotics*, 2005.

[3] M. Seiz, P. Jensfelt, and H. I Christensen, ”Active Exploration for Feature Based Global Localization,” *Proc. Of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol 1., pp. 281-287, 2000.

[4] P. Althaus and H. I. Christensen, ”Automatic Map Acquisition for Navigation in Domestic Environments,” *Proc. Of IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1151-1156, 2003.

[5] E. A. Topp and H. I. Christensen, ”Tracking for Following and Passing Persons,” *Proc. Of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2321-2327, 2005.

[6] E. A. Topp and H. I. Christensen, ”Topological Modelling for Humand Augmented Mapping,” *Proc. Of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2257-2263, 2006.

[7] A. Diosi, G. Taylor, and L. Kleeman, ”Interactive SLAM using Laser and Advanced Sonar,” *Proc. Of IEEE International Conference on Robotics and Automation*, pp. 1103-1108, 2005.

[8] Ju-Hong Park, Soohwan Kim, Nakju lett Doh, and Sung-Kee Park, ”Vision-based Global Localization Using a Hybrid Map Representation,” *Proc. Of International Conference on Control, Automation and Systems*, vol. 1, pp. 1104-1108, 2008.

[9] Y. Ke and R. Sukthankar, ”PCA-SIFT: A More Distinctive Representation for Local Image Descriptors,” *Proc. Of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 506-513, 2004.

[10] McNamara T. P., ”Mental Representations of Spatial Relations,” *Cognitive Psychology*, vol. 18, pp. 87-121, 1986.

[11] http://www.activrobots.com/SOFTWARE/aria.html